REMARKS

Claims 1-29 are pending. No claims have been amended. The specification has been amended to correct clerical errors with respect to reference numerals either appearing in the accompanying drawings or inadvertently mislabeled. No new matter has been entered.

Claims 1-6, 8-13, 15-18, 20-23, and 25-28 stand rejected under 35 U.S.C. 103(a) as being obvious over U.S. Patent application No. 6,131,163, issued to Wiegel, in view of U.S. Patent No. 6,279,113, issued to Vaidya. Applicant traverses the rejection. To establish a *prima facie* case of obviousness: (1) there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or combine the reference teachings; (2) there must be a reasonable expectation of success; and (3) the combined references must teach or suggest all the claim limitations. MPEP § 2143. A *prima facie* case of obviousness has not been shown.

For instance, Claim 1 defines a system for intrusion detection data collection using a protocol stack multiplexor. Claim 1 recites a protocol stack multiplexor collecting data directly from the protocol stack from at least one of the processed data packets. Claim 1 further recites an interface interfacing directly into at least one such protocol layer through redirected references to the data packet processing procedures comprised within the at least one such protocol layer. Claim 1 further recites a logical reference to the processed data packets obtained from the interfaced protocol layer, the logical reference referring to a memory block in the kernel memory space within which the processed data packets are stored and provided to an intrusion detection analyzer executing within user memory space.

For instance, Claim 8 defines a method for intrusion detection data collection using a protocol stack multiplexor. Claim 8 recites collecting data directly from the protocol stack from at least one of the processed data packets using a protocol stack multiplexor. Claim 8 further recites interfacing directly

into at least one such protocol layer through redirected references to the data packet processing procedures comprised within the at least one such protocol layer. Claim 8 further recites obtaining a logical reference to the processed data packets from the interfaced protocol layer, the logical reference referring to a memory block in the kernel memory space within which the processed data packets are stored. Claim 8 further recites providing the logical reference to an intrusion detection analyzer executing within user memory space.

For instance, Claim 15 defines a storage medium for intrusion detection data collection using a protocol stack multiplexor. Claim 15 recites collecting data directly from the protocol stack from at least one of the processed data packets using a protocol stack multiplexor. Claim 15 further recites interfacing directly into at least one such protocol layer through redirected references to the data packet processing procedures comprised within the at least one such protocol layer. Claim 15 further recites obtaining a logical reference to the processed data packets from the interfaced protocol layer, the logical reference referring to a memory block in the kernel memory space within which the processed data packets are stored. Claim 15 further recites providing the logical reference to an intrusion detection analyzer executing within user memory space.

For instance, Claim 20 defines a system for detecting network intrusions using a protocol stack multiplexor. Claim 20 recites a protocol stack multiplexor interfaced directly to at least one such protocol layer through a set of redirected pointers to the processing procedures of the interfaced protocol layer. Claim 20 further recites a data packet collector referencing at least one of the read queue and the write queue for the associated protocol layer. Claim 20 further recites a data packet exchanger communicating a memory reference to each transitory data packet from the referenced at least one of the read queue and the write queue for the associated protocol layer. Claim 20 further recites an analysis module receiving the communicated memory reference and performing intrusion detection based thereon.

For instance, Claim 25 defines method for detecting network intrusions using a protocol stack multiplexor. Claim 25 recites interfacing a protocol stack

multiplexor directly to at least one such protocol layer through a set of redirected pointers to the processing procedures of the interfaced protocol layer. Claim 25 further recites referencing at least one of the read queue and the write queue for the associated protocol layer. Claim 25 further recites communicating a memory reference to each transitory data packet from the referenced at least one of the read queue and the write queue for the associated protocol layer. Claim 25 further recites receiving the communicated memory reference into an analysis module and performing intrusion detection based thereon.

The Wiegel patent discloses a protocol stack proxy that is coupled between a device driver on a computer system configured to receive data from a network and one or more components of a network operating system (Abstract). The protocol stack proxy is coupled to a protocol proxy manager along a bi-directional communication path and the protocol stack proxy has one or more protocol proxy layers that include instructions that cause a processor to carry out security checks on packets, including accepting or rejecting a data packet based on a security policy criteria (Col. 8, lines 13-18 and 35-54; Col. 9, lines 39-42).

The protocol proxy manager intercepts packets before the packets reach the protocol stack and all devices attached to the system can only be accessed through the protocol proxy manager, which poses as a surrogate device driver (Col. 9, lines 13-18). Each protocol proxy layer is configured to operate in the same manner as the corresponding protocol layer in protocol stack (Col. 7, line 63-Col. 8, line 3 and Col. 8, lines 32-34). During packet processing, the protocol proxy manager stores state information in a set of state variables and the values in the state variables are passed to each of the protocol proxy layers, which then carry out the security checks on the packets (Col. 9, line 65-Col. 10, line 2).

The Vaidya patent discloses a signature-based dynamic network intrusion detection system (Abstract). Attack signature profiles descriptive of identifiable characteristics associated with particular network intrusion attempts are associated with network objects (Col. 3, lines 12-15). Data transmitted over a network is monitored by a data monitoring device to detect the data addressed to the network objects (Col. 3, lines 39-41). A processor processes at least one attack signature

profile to determine if data addressed to the network objects is associated with a network intrusion (Col. 3, lines 44-47). Data collectors are placed within the network to collect packets and each data collector only monitors a network segment on which the collector is located or a point of entry from an open network (Col. 3, lines 62-65; Col. 5, lines 5-26). The data collectors include reaction modules, which can terminate sessions, trace sessions or alert a network administrator of a suspected attack (Col. 6, lines 21-26; Col. 7, lines 44-45).

The combination of the Wiegel and Vaidya references fail to teach or suggest all the claim limitations of Claims 1, 8, 15, 20, and 25. In particular, there is no teaching or suggestion that discloses a protocol stack multiplexor collecting data directly from the protocol stack from at least one of the processed data packets, per Claims 1, 8 and 15, or a protocol stack multiplexor interfaced directly to at least one such protocol layer through a set of redirected pointers to the processing procedures of the interfaced protocol layer, per Claims 20 and 25. Rather, Wiegel teaches a protocol proxy manager that performs a redirection of data packets from an existing protocol stack and a proxy protocol layers that pose as surrogate protocol layers (Col. 8, line 66-Col 9, line 12). Vaidya teaches data collector monitors that collect complete packets, which perform *raw* packet-monitoring to ascertain the network address of each monitored packet (Col. 3, lines 56-58). Moreover, the reaction module in Vaidya reacts to packets, not collects packets (Col. 7, lines 44-45).

Moreover, there is no teaching or suggestion that discloses an interface interfacing directly into at least one such protocol layer through redirected references to the data packet processing procedures comprised within the at least one such protocol layer, per Claims 1, 8 and 15. Rather, Wiegel teaches a protocol proxy manager and protocol proxy layers that are executed directly, not through redirected references, and that *displace* the actual network stack by redirecting all packets away from the network stack (Col. 9, lines 13-18). Vaidya teaches data collectors and configuration builders that are also executed directly, not through redirected references, and that include reaction modules that function as active countermeasures to suspect packets by terminating or tracing sessions or

alerting a network administrator (Col. 6, lines 21-26; Col 7, lines 26-30).

In addition, there is no teaching or suggestion that discloses a logical reference to the processed data packets obtained from the interfaced protocol layer, the logical reference referring to a memory block in the kernel memory space within which the processed data packets are stored and provided to an intrusion detection analyzer executing within user memory space, per Claims 1, 8 and 15, or communicating a memory reference to each transitory data packet from the referenced at least one of the read queue and the write queue for the associated protocol layer and receiving the communicated memory reference into an analysis module and performing intrusion detection based thereon, per Claims 20 and 25. Rather, Wiegel teaches a protocol proxy manager that *stores* information into state variables and the *values* of the state variables are passed to the proxy protocol layers, rather than only by passing logical references to memory blocks (Col. 9, line 65-Col. 10, line 2). Vaidya teaches data collection monitors that monitor packets collected directly from a network and not from an interfaced protocol layer (Col. 5, lines 5-26). Moreover, Vaidya teaches processing the collected packets by extracting MAC header information, IP header information, transport header information, and application information from the data packets (Col. 18-24).

Moreover, the combination of the Wiegel and Vaidya references fails to provide some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify or combine the reference teachings. The base reference, Wiegel, teaches an network gateway mechanism that intercepts data packets by redirecting the data packets away from the actual network protocol stack using a protocol proxy manager. Wiegel further teaches accepting or rejecting the data packets based on security checks performed by the protocol proxy layers. Vaidya teaches a signature-based dynamic intrusion detection system that collects raw packets and processes each packet against a set of stored attack signature profiles. Thus, both Wiegel and Vaidya teach processing packets independently from the actual protocol stack and performing active security checks on the processed packets to respectively check

security or detect intrusion, rather than directly interfacing to the actual protocol layers, obtaining logical references to processed data packets and providing the logical references for use in intrusion detection, per Claims 1, 8, 15, 20, and 25

Finally, there would be no reasonable expectation of success. Wiegel and Vaidya teach away from Claims 1, 8, 15, 20, and 25. Wiegel and Vaidya both operate on actual data packets, rather than logical references to a memory block storing processed data packets. The approaches taken by Wiegel and Vaidya require that the original data packets be copied and stored, which is a solution that is not scaleable as the volume of packets and the speed at which the packets are received increases. Moreover, Wiegel and Vaidya are active counter measures that both directly perform some form of security or intrusion check on the data packets, rather than functioning as a collector of logical references to processed data packets for use by an intrusion detector. Therefore, there can be no expectation of success when both references teach away from the invention as claimed. A *prima facie* case of obviousness has not been shown for Claims 1, 8, 15, 20, and 25.

Claims 2-6 are dependent on Claim 1 and are patentable for the above-stated reasons, and as further distinguished by the limitations recited therein. Claims 9-13 are dependent on Claim 8 and are patentable for the above-stated reasons, and as further distinguished by the limitations recited therein. Claims 16-18 are dependent on Claim 15 and are patentable for the above-stated reasons, and as further distinguished by the limitations recited therein. Claims 21-23 are dependent on Claim 20 and are patentable for the above-stated reasons, and as further distinguished by the limitations recited therein. Claims 26-28 are dependent on Claim 25 and are patentable for the above-stated reasons, and as further distinguished by the limitations recited therein. Accordingly, as a *prima facie* case of obviousness has not been shown for Claims 1-6, 8-13, 15-18, 20-23, and 25-28, withdrawal of the rejection for obviousness under 35 U.S.C. 103(a) is requested.

Claims 8, 14, 19, 24, and 29 stand rejected under 35 U.S.C. 103(a) as being obvious over Wiegel, in view of Vaidya, and further in view of U.S. Patent

No. 6,489,666, issued to Shanklin et al. Applicant traverses the rejection. A *prima facie* case of obviousness has not been shown

Claim 14 is dependent on Claim 8 and is patentable for the above-stated reasons, and as further distinguished by the limitations recited therein. Claim 19 is dependent on Claim 15 and is patentable for the above-stated reasons, and as further distinguished by the limitations recited therein. Claim 24 dependent on Claim 20 and is patentable for the above-stated reasons, and as further distinguished by the limitations recited therein. Claim 29 is dependent on Claim 25 and is patentable for the above-stated reasons, and as further distinguished by the limitations recited therein. Accordingly, as a *prima facie* case of obviousness has not been shown for Claims 8, 14, 19, 24, and 29, withdrawal of the rejection for obviousness under 35 U.S.C. 103(a) is requested

The prior art made of record and not relied upon has been reviewed by the applicant and is considered to be no more pertinent than the prior art references already applied.

Claims 1-29 are believed to be in a condition for allowance. Entry of the foregoing amendments is requested and a Notice of Allowance is earnestly solicited. Please contact the undersigned at (206) 381-3900 regarding any questions or concerns associated with the present matter.

Respectfully submitted,

Dated: June 11, 2004

By: _____
Patrick J.S. Inouye, Esq.
Reg. No. 40,297

Law Offices of Patrick J.S. Inouye
810 Third Ave, Suite 258
Seattle, WA 98104

Telephone: (206) 381-3900
Facsimile: (206) 381-3999